Power of the interactive proof systems with verifiers modeled by semi-quantum two-way finite automata

Shenggen Zheng[1,*], Jozef Gruska[1,†], Daowen Qiu[2,3,‡]

[1] *Faculty of Informatics, Masaryk University, Brno 60200, Czech Republic*

[2] *Department of Computer Science, Sun Yat-sen University, Guangzhou 510006, China*

[3] *SQIG–Instituto de Telecomunicações, Departamento de Matemática, Instituto Superior Técnico, Technical University of Lisbon, Av. Rovisco Pais 1049-001, Lisbon, Portugal*

Abstract

Interactive proof systems (IP) are very powerful – languages they can accept form exactly PSPACE. They represent also one of the very fundamental concepts of theoretical computing and a model of computation by interactions. One of the key players in IP are verifiers. In the original model of IP whose power is that of PSPACE, the only restriction on verifiers is that they work in randomized polynomial time. Because of such key importance of IP, it is of large interest to find out how powerful will IP be when verifiers are more restricted. So far this was explored for the case that verifiers are *two-way probabilistic finite automata* (Dwork and Stockmayer, 1990) and *one-way quantum finite automata* as well as *two-way quantum finite automata* (Nishimura and Yamakami, 2009). IP in which verifiers uses *public randomization* is called *Arthur-Merlin proof systems* (AM). AM with verifiers modeled by Turing Machines augmented with a fixed-size quantum register (qAM) were studied also by Yakaryilmaz (2012). He proved, for example, that an NP-complete language $L_{knapsack}$, modeling the 0 - 1 knapsack problem, can be recognized by a qAM whose verifier is a *two-way finite automaton* working on quantum mixed states using superoperators.

In this paper we explore the power of AM for the case that verifiers are *two-way finite automata with quantum and classical states* (2QCFA) – introduced by Ambainis and Watrous in 2002 – and the communications are classical. It is of interest to consider AM with such "semi-quantum" verifiers because they use only limited quantum resources. Our main result is that such Quantum Arthur-Merlin proof systems (QAM(2QCFA)) with polynomial expected running time are more powerful than in the case verifiers are two-way probabilistic finite automata (AM(2PFA)) with polynomial expected running time. Moreover, we prove that there is a language which can be recognized by an exponential expected running time QAM(2QCFA), but can not be recognized by any AM(2PFA), and that the NP-complete language $L_{knapsack}$ can also be recognized by a QAM(2QCFA) working only on quantum pure states using unitary operators.

Keywords: Quantum computing; quantum finite automata; quantum Arthur-Merlin proof system;

[*] *E-mail address:* zhengshenggen@gmail.com (S. Zheng)

[†] *E-mail address:* gruska@fi.muni.cz (J. Gruska)

[‡] Corresponding author. *E-mail address:* issqdw@mail.sysu.edu.cn (D. Qiu)

two-way finite automata with quantum and classical states.

1 Introduction

An important way to get deeper insights into the power of various quantum resources and operations is to explore the power of various quantum variations of the basic models of classical automata. Of a special interest is to do that for various quantum variations of the classical finite automata, especially for those that use limited amounts of always expensive quantumness – quantum resources: states, correlations, operations and measurements. This paper aims to contribute to such a line of research.

There are two basic approaches how to introduce quantum features to classical models of finite automata. The first one is to consider quantum variants of the classical *one-way (deterministic) finite automata* (1FA or 1DFA) and the second one is to consider quantum variants of the classical *two-way finite automata* (2FA or 2DFA). Already the very first attempts to introduce such models, by Moore and Crutchfields [21] as well as Kondacs and Watrous [17] demonstrated that in spite of the fact that in the classical case, 1FA and 2FA have the same recognition power, this is not so for their quantum variations (in case only unitary operations and projective measurements are considered as quantum operations). Moreover, already the first model of *two-way quantum finite automata* (2QFA), namely that introduced by Kondacs and Watrous, demonstrated that quantum variants of 2FA are much too powerful – they can recognize even some *non-context free languages* and are actually not really finite in a strong sense [17]. It started to be therefore of interest to introduce and explore some "less quantum" variations of 2FA and their power [1, 2, 3, 6, 5, 18, 19, 20, 30, 31].

A "hybrid" quantum variations of 2FA, namely, *two-way finite automata with quantum and classical states* (2QCFA) were introduced by Ambainis and Watrous [3]. Using this model they were able to show, in an elegant way, that already an addition of a single qubit to a classical model can much increase its power. A 2QCFA is essentially a classical 2FA augmented with a quantum memory of constant size (for states of a fixed Hilbert space) that does not depend on the size of the (classical) input. In spite of such a restriction, 2QCFA have been shown to be more powerful than *two-way probabilistic finite automata* (2PFA) [3, 34, 35].

In mid 1980s, Babai [4] and Goldwaser et al. [12], independently, introduced so-called *interactive proof systems* with unlimited power provers and polynomial power randomized verifiers. A famous result of [27], stated as IP=PSPACE, that languages recognized by IP are exactly those from PSPACE, demonstrated enormous power hidden in simple interactions of IP.

It is therefore natural to explore power also of some weaker variations of IP. Since unlimited power of provers seems to be very essential for the whole concept of IP, the research started to focus on the cases with limited power verifiers. This has been done at first by Dwork and Stockmeyer [9] – they explored the case that verifiers are *two-way probabilistic finite automata* (IP(2PFA)). They showed that every language in the class EXP can be accepted by some IP(2PFA). However, the set of languages recognized by such IP in which verifiers use *public randomization* (also called Arthur-Merlin proof systems) is a proper subset of P. Later, Nishimura and Yamakami [23] explored the case that verifiers are modeled by *one-way quantum finite automata* as well as *two-way quantum finite automata* and demonstrated strengths and weaknesses of both IP.

Of importance is also a variant of IP, called *Arthur-Merlin proof systems* (AM). The difference between IP and AM is that the prover of IP has at each step only *partial information* of the configuration of the verifier while the prover of AM always has *complete information* of the current configuration of the verifier. Also for such *interactive proof systems* it is of importance to explore their power for the case that verifiers have a more limited power and to find out relations between IP and AM with verifiers of different power. AM with verifiers modeled by Turing Machines augmented with a fixed-size quantum register (qAM) were studied also in [33, 32] and it was shown that the an NP-complete language $L_{knapsack}$, modeling the 0 - 1 knapsack problem, can be recognized by a qAM whose verifier is a *two-way finite automaton* working on quantum mixed states using superoperators. In Yakaryilmaz's notation, *two-way finite automata* working on quantum mixed states using superoperators are called 2QCFA. However, 2QCFA as defined originally in [3], are working only on quantum pure states using unitary operators. They can be simulated efficiently by *two-way finite automata* working on quantum mixed states, but whether *two-way finite automata* working on quantum mixed states can be simulated by 2QCFA, or not, is unknown. The model of 2QCFA we use is that of [3] and it is weaker, actually a special case, of the model used in [33, 32]. Our results concerning the acceptance of the language $L_{knapsack}$ are therefore stronger.

Our model will be denoted as QAM(2QCFA). One can see this model also as a classical AM augmented with a quantum memory of constant size – to store quantum states of a fixed Hilbert space – that does not depend on the size of the (classical) input. Our main results show that such models are more powerful than AM(2PFA) – that is AM with 2PFA as verifiers, and the NP-complete language can be recognized by QAM(2QCFA).

The paper is structured as follows. In Section 2 all models involved are described in detail. After that we show for the language $L_{middle} = \{xay \,|\, x, y \in \{a, b\}^*, |x| = |y|\}$ that for any $0 < \varepsilon < 1/2$ there is a QAM(2QCFA) $A(P, V_\varepsilon)$ – with the prover P and the verifier V_ε that accepts L_{middle} with one-sided error ε in a polynomial expected running time – notation QAM(ptime-2QCFA). This language cannot be recognized by any AM(2PFA) in polynomial expected running time, as shown in [9]. As the next we show, for the language $L_{mpal} = \{xax^R \,|\, x \in \{a, b\}^*\}$, that for any $0 < \varepsilon < 1/2$ there is a QAM(2QCFA) $A(P, V_\varepsilon)$ that can recognize L_{mpal} with one-sided error ε in an exponential expected running time. We will prove that this language cannot be recognized at all by an AM(2PFA). These results show that QAM(2QCFA) are more powerful than AM(2PFA). Afterwards we show that there is an NP-complete language, namely $L_{knapsack}$, modeling the 0 - 1 knapsack problem, that can be recognized by QAM(2QCFA) in an exponential expected running time. Finally, we discuss languages, $L_1 = \{w \,|\, \exists s, t, u, v \in \{a, b\}^*,\ w = sbt = ubv, |s| = |v|\}$ and $L_2 = \{w \,|\, \exists s, t, u, v \in \{a, b\}^*,\ w = sat = ubv, |s| = |v|\}$, that can be recognized by QAM(ptime-2QCFA). The language L_1 is proved to be nonstochastic. The language L_2, the set of nonpalindromes, is stochastic [10].

The fact that the *non-regular language* L_{middle} can be recognized by a QAM(ptime-2QCFA) and it seems that it can not be recognized by a 2QCFA, indicates that QAM(ptime-2QCFA) are likely more powerful than 2QCFA. Interesting enough, this situation seems to be different for 2PFA. It is still an open problem to find out whether there is a *non-regular language* that can be recognized by AM(ptime-2PFA), but we know that any 2PFA needs exponential time to recognize a *non-regular language* [8, 13] .

2 Basic models

At first we introduce formally the model IP(2PFA) and afterwards also the model QIP(2QCFA). Concerning basics of quantum computation we refer the reader to [14, 22], and concerning basics of classical and quantum automata we refer the reader to [14, 15, 16, 24, 26].

2.1 Model IP(2PFA)

Notation: A *coin-tossing distribution* on a finite set S is a mapping $\phi : S \to \{0, 1/2, 1\}$ such that $\sum_{s \in S} \phi(s) = 1$.

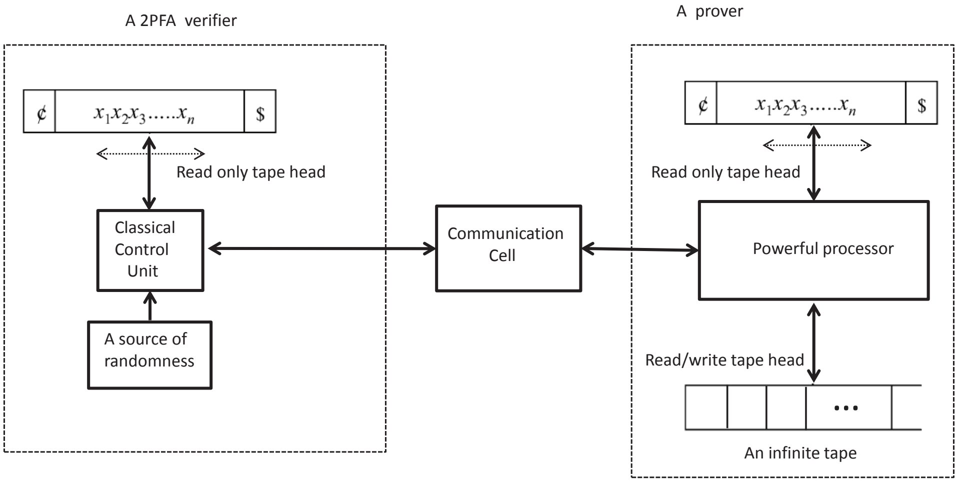

Figure 1: A schematic model of an IP(2PFA)

Definition 1. An IP(2PFA) $A = (P, V)$, where P is a so called *prover* and V is a so called *verifier* (that are specified bellow and communicate through a communication cell as illustrated in Figure 1).

An action of A starts with a step of the verifier. The verifier's head scans the left end-marker, and the verifier writes a symbol to the communication cell. That is followed by an action of the prover who writes its response into the communication cell. Such rounds of steps continue until the verifier decides to end it.

The verifier V is a 2PFA specified as follows: $V = (S, \Sigma, \delta, s_0, S_{acc}, S_{rej})$, where

1. S is a finite set (of classical states) partitioned into subsets R, C, H of reading, communication and halting states, respectively.

2. $s_0 \in R$ is the initial reading state.

3. H is partitioned into sets S_{acc} and S_{rej} of the accepting and rejecting states.

4. Σ is a finite input alphabet that is extended into the alphabet $\Sigma' = \Sigma \cup \{ ¢, \$\}$, where $¢ \notin \Sigma$ will be used as the left end-marker and $\$ \notin \Sigma$ will be used as the right end-marker. Γ is a communication alphabet (shared by both the verifier and the prover).

5. δ is a transition mapping defined as follows:

   (a) For each reading state $s \in R$ and $\sigma \in \Sigma'$, $\delta(s, \sigma)$ is a coin-tossing distribution on $S \times \{-1, 0, 1\}$, where $-1, 0$ and 1 specify the tape head move – one cell left, no move, one cell right. It is assumed that δ is well defined in the sense that on end-markers it never determines a move outside of the tape region separated by end-markers ¢ and \$. Therefore δ specifies, for each state of the control unit and each symbol on the tape, the probability that the control unit will be in a particular new state and that the tape head moves in a particular direction.

   (b) With each communication state $s \in C$, a unique communication symbol γ_s is associated. In the case the verifier is in such a communication state s, then it writes γ_s in the communication cell and then the prover gets into an action. It reads the symbol stored in the communication cell and, depending on the whole communication history and the input string (that is the same for both the prover and the verifier), the prover writes a symbol $\gamma \in \Gamma$ into the communication cell. Afterwards the verifier gets into the action that depends on its current state and on the symbol in the communication cell. For each pair $s \in S$ and $\gamma \in \Gamma$ a coin-tossing distribution function on the set S is defined that determines the probability for each state from S that it should be the next state of the verifier. At such after-communication step the head of the verifier does not move.

   (c) The verifier halts when it gets into a halting state. Therefore there is no need to define δ on states in H.

The prover P is all-powerful and at each communication step what the prover writes into the communication cell depends on the whole input and the whole communication history up to that point of communication. Namely, it is determined, for an input string x and communication history $y \in \Gamma^*$ to that point, by a coin-tossing distribution $\rho(x, y)$ defined on Γ.

Note. The communication cell always holds a symbol from the communication alphabet Γ. Whenever the verifier needs some information from the prover, the verifier writes a request to the communication cell via a symbol from the alphabet Γ and the prover responds. The verifier then continues, probabilistically, depending on the prover's respond.

Remark 1. In the above definition, the prover's understanding of the verifier's computation is only through the communication cell that can contain information only from the finite set Γ. This mode of communication is called *private coins* [11] or *partial information* [7]. One can consider also IP in which the prover has complete information on the current configuration of the verifier. Such a communication mode is called *public coins* [11] or *complete information* [7] mode. In Babai's terminology, an IP with the public coins communication mode is called Arthur-Merlin proof system (AM). In an AM(2PFA), the verifier in each step automatically sends its configuration information, that is the communication symbol corresponding to the communication alphabet contains an element of $S \times \{-1, 0, 1\}$, that is, the current state and the last move of the head, to the prover through the communication cell in every computation step. This information is sufficient for the all-powerful prover to keep the track of the configuration of the verifier because the prover knows the strategy of the verifier.

The computation of (P, V) on an input string w starts with the string ¢w\$ on the input tapes of both the verifier and the prover. The tape head of the verifier is positioned on the left end-marker ¢

and the verifier begins to act with the initial state s_0 in its control unit. The action of the verifier and the prover in the next steps is then governed (probabilistically) by the transition functions δ and ρ, as defined above, until the verifier enters a halting state. For a particular input w and a halting state s let $Pr_{(P,V)}(w,s)$ be the probability that IP (P,V) halts its computation on w in the state s. The probability that the verifier halts in a halting state s on the input w is taken over all random choices of the verifier and the prover. If s is in an accepting (a rejecting) state, then the input is accepted (rejected).

The prover-verifier pair (P, V_ε) is an AM(2PFA) for (accepting) a language $L \subset \Sigma^*$ with an error probability $\varepsilon < 1/2$ if

1. **(Completeness condition)**: for all $w \in L$, $Pr[(P, V_\varepsilon)$ accepts $w] \geq 1 - \varepsilon$, and
2. **(Soundness condition)**: for all $w \notin L$ and any prover P^*, $Pr[(P^*, V_\varepsilon)$ rejects $w] \geq 1 - \varepsilon$.

We say that a language L is recognized by AM(2PFA) if for some $\varepsilon < 1/2$, there is an AM(2PFA) (P, V_ε) that accepts the language L with the error probability ε.

2.2 Model QIP(2QCFA)

2QCFA were introduced by Ambainis and Watrous [3] and explored by Qiu, Yakaryilmaz and others [25, 30, 34, 35]. Informally, a 2QCFA can be seen as a 2DFA with an access to a quantum memory for states of a fixed Hilbert space upon which at each step either a unitary operation is performed or a projective measurement and the outcomes of which then probabilistically determines the next move of the underlying 2DFA.

A quantum analogue of IP was introduced by Watrous in 2003 [28] and since then the study of QIP has become an interesting and important topic of quantum complexity theory. In particular, quantum analogues of Babai's *Arthur-Merlin proof system* called now *quantum Arthur-Merlin proof system* (QAM) have drawn a significant attention [29, 32, 33]. In [32, 33], for Arthur-Merlin IP verifiers are augmented by a fixed-size quantum memory. In [23] *one-way quantum finite automata* (both of the *measure-many* and *measure-once* types) and *two-way quantum finite automata* are considered as verifiers.

In this paper, such IP are mainly considered in which verifiers are not 2PFA, as in [23], but 2QCFA and the verifier and the prover have just classical communication. More exactly, verifiers are augmented by a quantum memory size of which does not depend on the size of the input. The formal definition is as follows:

Definition 2. A QIP(2QCFA) is given by a pair (P,V), where P is a prover and V is a verifier, as illustrated in Figure 2, and the prover communicate classically with the verifier through a special communication cell.

In such a QIP(2QCFA) (P,V), the verifier V is a 2QCFA $A = (Q, S, \Sigma, \Gamma, \Theta, \delta, |q_0\rangle, s_0, S_{acc}, S_{rej})$, where

1. Q is a finite set of an n basic orthonormal quantum states.

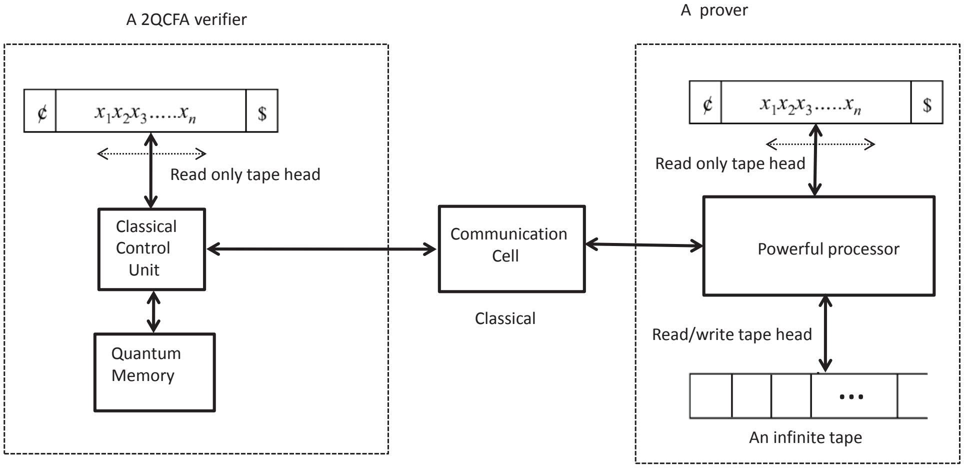

Figure 2: A schematic model of a QIP(2QCFA)

2. S is a finite set of classical states that is partitioned into subsets R, C, H of the reading, communication and halting states..

3. Σ is a finite set of input symbols extended to the tape symbols $\Sigma' = \Sigma \cup \{¢, \$\}$, where ¢ is used as the left end-marker and \$ is used as the right end-marker.

4. Γ is a finite set of communication symbols.

5. $|q_0\rangle \in Q$ is the initial quantum state.

6. s_0 is the initial classical state.

7. The set H is partitioned into subsets S_{acc} and S_{rej} of the accepting and rejecting states.

8. Θ is a mapping $\Theta : R \times \Sigma' \to U(H(Q)) \cup O(H(Q))$, where U(H(Q)) and O(H(Q)) are sets of unitary operations and projective measurements on the Hilbert space generated by quantum states from Q.

9. δ is a classical transition function defined as follows.

   (a) For each reading state $s \in R$ and a tape symbol σ

   i. If $\Theta(s, \sigma) \in U(H(Q))$, then the unitary operation $\Theta(s, \sigma)$ is applied on the current state of quantum memory to produce a new quantum state and, in addition, if

$$\delta : R \times \Sigma' \to S \times \{-1, 0, 1\},$$

   then, for the case $\delta(s, \sigma) = (s', d)$, the new classical state of the verifier is s' and its head moves in the direction d.

   ii. If $\Theta(s, \gamma) \in O(H(Q))$, then $\Theta(s, \sigma)$ is a projective measurement with a set of possible eigenvalues $E_{\Theta(s,\lambda)} = \{\lambda_1, \ldots, \lambda_n\}$ and projectors $\{P_1, \ldots, P_n\}$, where P_i is the projector onto the eigenspaces generated by eigenvectors corresponding to λ_i. In such a case

$$\delta_{(s,\sigma)} : E \to S \times \{-1, 0, 1\}$$

and $\delta_{(s,\sigma)}(\lambda_i) = (s', d)$ means that when the projective measurement outcome is λ_i, then the new classical state is s' and its head moves in the direction d.

(b) With each communicating state $s \in C$ a communication symbol γ_s is associated. For each $s \in C$ and $\gamma \in \Gamma$, $\delta(s, \gamma) \in S$. This has the following meaning: If the verifier gets into a state $s \in C$, it writes γ_s into the communication cell and then the prover, depending on the whole communication history and on the input w writes a $\gamma \in \Gamma$ into the communicating cell and the verifier comes into the action. If $\delta(s, \gamma) = s'$, then s' is the new state of the verifier and the tape head of the verifier does not move.

(c) The verifier halts and accepts (rejects) the input when it enters a classical accepting (rejecting) state from H.

The prover is a processor of an unlimited power as in the general case of IP.

Remark 2. A QIP(2QCFA) is a QAM(2QCFA) if the verifier sends, at the end of each its communication round through a communication symbol information about its current configuration, that is an element of $S \times \{-1, 0, 1\} \times E$, where E is the set of all possible measurement outcomes, that is the communication symbol contains the current classical state, the last move of the head and the projective measurement outcome (let a special symbol is used if there was no measurement in the computation step), to the prover through the communication cell in every computation step. This information is sufficient for the all-powerful prover to keep the track of the configuration of the verifier because the prover knows the strategy of the verifier.

The computation of QIP(2QCFA) with a prover P and a verifier V on an input $w \in \Sigma^*$ starts with the string $¢x\$$ on the input tapes of both of them. At the start, the tape head of the verifier is positioned on the left end-marker and the verifier begins the computation in the classical initial state and in the initial quantum state. In each of the next steps, if the current classical state of the verifier is s and the current quantum state of the verifier is $|\psi\rangle$ and the scanning symbol is σ, then quantum and classical states are changed according to $\Theta(s, \sigma)$ and δ as follows.

1. If $s \in R$, then:

   (a) If $\Theta(s, \sigma)$ is a unitary operation U, then U is applied on the current quantum state $|\psi\rangle$, changing it into the quantum state $U|\psi\rangle$, and $\delta(s, \sigma) = (s', d)$ makes s' to be a new classical state and the head moves in the direction d. If $s \in S_{acc}$ ($s \in S_{rej}$), then the input is accepted (rejected).

   (b) If $\Theta(s, \sigma)$ is a projective measurement, then the current quantum state is changed to $P_j|\psi\rangle / ||P_j|\psi\rangle||$ with the probability $||P_j|\psi\rangle||^2$ and in such a case $\delta_{(s,\sigma)}$ is a mapping from the set of potential classical outcomes (eigenvalues) of the measurement to $S \times \{-1, 0, 1\}$. In particular, if the measurement coutcome is λ_i and $\delta_{(s,\sigma)}(\lambda_i) = (s', d)$, then:

      i. if $s' \in S - H$, then s' is the new classical state and the head moves in the direction d;

      ii. if $s' \in S_{acc}$ ($s' \in S_{rej}$), then the verifier accepts (rejects) the input and computation halts.

2. If the current classical state $s \in C$, then the verifier sends γ_s to the prover through the communication cell. Depending on this and all previously obtained symbols from the verifier, as well

as on the input string, the prover sends to the communication cell a symbol γ. If $\delta(s,\gamma) = s'$, then s' will become the new classical state of the verifier and the input head does not move. In case $s' \in S_{acc}$ ($s' \in S_{rej}$) the input is accepted.

The probability that an input world is accepted is defined in a similar way as in the case of 2QCFA.

The prover-verifier pair (P, V_ε) is a QAM(2QCFA) for (accepting) a language $L \subset \Sigma^*$ with one-sided error $0 < \varepsilon < 1/2$ if

1. **Completeness condition**: for all $w \in L$, $Pr[(P, V_\varepsilon)$ accepts $w] = 1$, and
2. **Soundness condition**: for all $w \notin L$ and any prover P^*, $Pr[(P^*, V_\varepsilon)$ rejects $w] \geq 1 - \varepsilon$.

We say that a language L is recognized by QAM(2QCFA) if for some $\varepsilon < 1/2$ there is a QAM(2QCFA) (P, V_ε) that accepts the language L with one-sided error ε.

3 Examples of languages recognized QAM(2QCFA)

In this section we provide a detailed proof for several languages that they are accepted by QAM(2QCFA).

3.1 Recognition of the language L_{middle}

The importance of the fact that the language L_{middle} can be recognized by a QAM(ptime-2QCFA) is underlined by the fact that there is no AM(ptime-2PFA) for this language [9].

Theorem 1. *For any $\varepsilon < 1/2$ there exists a QAM(2QCFA) A_ε with a verifier-prover pair (P, V_ε) for the language L_{middle} with one-sided error ε in the expected running time $\boldsymbol{O}\left(\frac{1}{\varepsilon}\left(n^4 + n^2 \log \frac{1}{\varepsilon}\right)\right)$, where n is the length of the input.*

```
Repeat the following steps ad infinity:
  1. Move the tape head to the symbol that is next to the left end-marker.
  2. Ask the prover whether the currently read symbol is the middle one.
     2.1 If the answer is "no", move the head to the right, apply transformation $U_\alpha$ and repeat Step 2.
     2.2 If the answer is "yes" and the scanned symbol is not $a$ the verifier rejects the input; otherwise
         the verifier moves the head one cell to the right and proceeds according to Step 3.
  3. Until the scanned symbol is the right end-marker, apply $U_{-\alpha}$ to the current quantum state and
     move the head one cell to the right.
4.0 When the right end-marker is reached, measure the quantum state in the basis $\{|q_0\rangle, |q_1\rangle\}$.
     4.1 If the result is $|q_1\rangle$, the input is rejected.
     4.2 Otherwise repeat the following subroutine two times:
         4.2.1 Move the head to the first symbol right to the left end-marker.
         4.2.2 Until the currently read symbol is one of the end-markers simulate a coin-flip and move
               the head right (left) if the outcome of the coin-flip is "head" ("tail").
  5. If the above process ends both times at the right end-marker, simulate $k$ coin-flips and if all outcomes
     are "head", then accepts the input.
```

Figure 3: Description of the behavior of the verifier V_ε in a QAM recognizing the language L_{middle}. The choice of k depends on ϵ, as discussed in the text.

Proof. At first we present informally the main idea of the proof. A 2QCFA verifier V_ε uses quantum memory with states generated by two orthogonal quantum states $|q_0\rangle$ and $|q_1\rangle$, where $|q_0\rangle$ will be the initial quantum state. At the beginning the input is shared by both the verifier and the prover. In the first part of the interaction/computation process, until the middle symbol is reached, for each input symbol σ the verifier does the following.

It applies the following unitary transformation U_α, $\alpha = \sqrt{2}\pi$,

$$U_\alpha|q_0\rangle = \cos\alpha|q_0\rangle + \sin\alpha|q_1\rangle, \quad U_\alpha|q_1\rangle = -\sin\alpha|q_0\rangle + \cos\alpha|q_1\rangle, \tag{1}$$

to the current quantum state (that is the state is rotated by the angle α) and asks the prover whether the next symbol is the middle one. The all-powerful prover provides the answer. If the next symbol is not the middle one, the verifier moves its head one cell to the right and this process repeats. When the symbol in the middle is reached, the verifier checks whether it is the symbol a. If not, the verifier rejects the input; otherwise it continues to read the input, symbol by symbol, and each time it applies the unitary transformation $U_{-\alpha}$ to the current quantum state (and therefore this quantum state is rotated by the angle $-\alpha$) until the right end-marker is reached. (During this process the verifier has no need to get any information from the prover.)

When the right end-marker is reached, the verifier measures the current quantum state in the basis $\{|q_0\rangle, |q_1\rangle\}$. If $|q_1\rangle$ is the resulting quantum state, the input is rejected, otherwise the verifier proceeds as shown in Figure 3.

Lemma 2. *If the input $w \in L_{middle}$, then the quantum state of the verifier V_ε will evolve with certainty into $|q_0\rangle$ when the right end-marker is reached in Step 4.*

Proof. If $w \in L_{middle}$, then there are strings $x, y \in \Sigma^*$ such that $|x| = |y|$ and $w = xay$. Since the all-powerful prover can tell the verifier for sure when the middle symbol is reached, the quantum state when the right end-marker is reached is

$$|q\rangle = (U_{-\alpha})^{|y|}(U_\alpha)^{|x|}|q_0\rangle = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}^{|y|} \begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix}^{|x|} |q_0\rangle \tag{2}$$

$$= \begin{pmatrix} \cos|y|\alpha & -\sin|y|\alpha \\ \sin|y|\alpha & \cos|y|\alpha \end{pmatrix} \begin{pmatrix} \cos|x|\alpha & \sin|x|\alpha \\ -\sin|x|\alpha & \cos|x|\alpha \end{pmatrix} |q_0\rangle = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} |q_0\rangle = |q_0\rangle. \tag{3}$$

□

Lemma 3. *[35] A coin flipping can be simulated by the verifier V_ε using states $|q_0\rangle$ and $|q_1\rangle$.*

Lemma 4. *[3] If the input $w \in L_{middle}$, then the execution of loops in the steps 4.2 and 5 leads to the acceptance with the probability $\frac{1}{2^k(n+1)^2}$.*

It may happen that the process described in Figure 3 does not terminate. This happens only if the result of the last coin-flip is "tail" and this happens either when the left end-marker is reached in Step 4.2 or this happens in Step 5. In such a case the quantum state of the verifier is $|q_0\rangle$ and a new iteration of the process starts.

Completeness condition: If $w \in L_{middle}$, then the quantum state of the verifier after Step 4.0 is $|q_0\rangle$ and the input is never rejected in Step 4.1. After the rest of the steps, in 4.2 and 5, the probability of accepting the input is, if we denote $n = |w|$, $P_a = 1/(2^k(n+1)^2)$ according to the previous lemma, and the probability of rejecting the input is $P_r = 0$. If the whole process is repeated for infinitum, the accepting probability is

$$Pr[(P^*, V_\varepsilon) \text{ rejects } w] = \sum_{i\geq 0}(1-P_a)^i(1-P_r)^iP_a = \frac{P_a}{P_a+P_r-P_aP_r} = \frac{P_a}{P_a} = 1. \tag{4}$$

Soundness condition: Let the input string $w \notin L_{middle}$ and $n = |w|$. Observe that the verifier, in its communication with the prover, waits only for the information that the currently read symbol is in the middle and let this comes after reading $0 < m < n$ symbols in the input string (this m can be different at different iterations and for different provers).

Lemma 5. *If $w \notin L_{middle}$, then the verifier rejects the input for any prover after Step 4.1 with probability at least $1/(2n^2+1)$.*

Proof. Suppose that a prover P^*, after m steps, informs the verifier that the next symbol is the middle one. If w does not have the form xay, $|x| = m$, then w is rejected in Step 2.2.

If $w = xay$ with $|x| = m$, then, before the measurement in Step 4.0 the quantum state of the verifier will be

$$|q\rangle = (U_{-\alpha})^{n-m-1}(U_\alpha)^m|q_0\rangle = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}^{n-m-1}\begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix}^{m}|q_0\rangle \tag{5}$$

$$= \begin{pmatrix} \cos(n-m-1)\alpha & -\sin(n-m-1)\alpha \\ \sin(n-m-1)\alpha & \cos(n-m-1)\alpha \end{pmatrix}\begin{pmatrix} \cos m\alpha & \sin m\alpha \\ -\sin m\alpha & \cos m\alpha \end{pmatrix}|q_0\rangle \tag{6}$$

$$= \begin{pmatrix} \cos(n-2m-1)\alpha & \sin(n-2m-1)\alpha \\ \sin(n-2m-1)\alpha & \cos(n-2m-1)\alpha \end{pmatrix}|q_0\rangle \tag{7}$$

$$= \cos\left((n-2m-1)\alpha\right)|q_0\rangle + \sin\left((n-2m-1)\alpha\right)|q_1\rangle. \tag{8}$$

The probability of observing $|q_1\rangle$ is $\sin^2\left(\sqrt{2}(n-2m-1)\pi\right)$ in Step 4.0. Without a loss of generality, we assume that $n-2m-1 > 0$. Let l be the closest integer to $\sqrt{2}(n-2m-1)$. If $\sqrt{2}(n-2m-1) > l$, then $2(n-2m-1)^2 > l^2$. So we get $2(n-2m-1)^2 - 1 \geq l^2$ and $l \leq \sqrt{2(n-2m-1)^2-1}$. Therefore

$$\sqrt{2}(n-2m-1) - l \geq \sqrt{2}(n-2m-1) - \sqrt{2(n-2m-1)^2-1} \tag{9}$$

$$= \frac{(\sqrt{2}(n-2m-1) - \sqrt{2(n-2m-1)^2-1})(\sqrt{2}(n-2m-1) + \sqrt{2(n-2m-1)^2-1})}{\sqrt{2}(n-2m-1) + \sqrt{2(n-2m-1)^2-1}} \tag{10}$$

$$= \frac{1}{\sqrt{2}(n-2m-1) + \sqrt{2(n-2m-1)^2-1}} > \frac{1}{2\sqrt{2}(n-2m-1)}. \tag{11}$$

Because l is the closest integer to $\sqrt{2}(n-2m-1)$, we have $0 < \sqrt{2}(n-2m-1) - l < 1/2$. Let $f(x) = sin(x\pi) - 2x$. We have $f''(x) = -\pi^2\sin(x\pi) \leq 0$ when $x \in [0, 1/2]$. That is to say, $f(x)$ is concave in the interval $[0, 1/2]$, and we have $f(0) = f(1/2) = 0$. So, for any $x \in [0, 1/2]$, $f(x) \geq 0$, that is $\sin(x\pi) \geq 2x$. Therefore, we have

$$\sin^2(\sqrt{2}(n-2m-1)\pi) = \sin^2\left((\sqrt{2}(n-2m-1) - l)\pi\right) \tag{12}$$

$$\geq \left(2(\sqrt{2}(n-2m-1)-l)\right)^2 = 4\left(\sqrt{2}(n-2m-1)-l\right)^2 \tag{13}$$

$$> 4\left(\frac{1}{2\sqrt{2}(n-2m-1)}\right)^2 = \frac{1}{2(n-2m-1)^2} > \frac{1}{2(n-2m-1)^2+1}. \tag{14}$$

If $\sqrt{2}(n-2m-1) < l$, then $2(n-2m-1)^2 < l^2$. So we get $2(n-2m-1)^2+1 \leq l^2$ and $l \geq \sqrt{2(n-2m-1)^2+1}$. We have therefore

$$\sqrt{2}(n-2m-1) - l \leq \sqrt{2}(n-2m-1) - \sqrt{2(n-2m-1)^2+1} \tag{15}$$

$$= \frac{\left(\sqrt{2}(n-2m-1) - \sqrt{2(n-2m-1)^2+1}\right)\left(\sqrt{2}(n-2m-1) + \sqrt{2(n-2m-1)^2+1}\right)}{\sqrt{2}(n-2m-1) + \sqrt{2(n-2m-1)^2+1}} \tag{16}$$

$$= \frac{-1}{\sqrt{2}(n-2m-1) + \sqrt{2(n-2m-1)^2+1}} < \frac{-1}{2\sqrt{2(n-2m-1)^2+1}} \tag{17}$$

and this implies that

$$l - \sqrt{2}(n-2m-1) > \frac{1}{2\sqrt{2(n-2m-1)^2+1}}. \tag{18}$$

Because l is the closest integer to $\sqrt{2}(n-2m-1)$, we have $0 < l - \sqrt{2}(n-2m-1) < 1/2$. Therefore,

$$\sin^2\left(\sqrt{2}(n-2m-1)\pi\right) = \sin^2\left((\sqrt{2}(n-2m-1)-l)\pi\right) \tag{19}$$

$$= \sin^2\left((l-\sqrt{2}(n-2m-1))\pi\right) \geq \left(2(l-\sqrt{2}(n-2m-1))\right)^2 \tag{20}$$

$$= 4\left(l-\sqrt{2}(n-2m-1)\right)^2 > 4\left(\frac{1}{2\sqrt{2(n-2m-1)^2+1}}\right)^2 = \frac{1}{2(n-2m-1)^2+1}. \tag{21}$$

As $0 < m < n$, we have $|n-2m-1| < n$, and therefore

$$\frac{1}{2(n-2m-1)^2+1} > \frac{1}{2n^2+1}. \tag{22}$$

So the lemma has been proved. □

If $w \notin L_{middle}$, then for any prover P^* the above verifier rejects the input after Step 4.1 with the probability

$$P_r > \frac{1}{2n^2+1} \tag{23}$$

according to Lemma 5. Moreover, after the last two steps the verifier accepts w with the probability

$$P_a = \frac{1}{2^k(n+1)^2}. \tag{24}$$

If $k = 1 + \lceil \log_2 1/\varepsilon \rceil$, then $\varepsilon \geq 1/2^{k-1}$.

If the whole process is repeated indefinitely, then the probability that the verifier rejects the input w for any prover P^* is

$$Pr[(P^*, V_\varepsilon) \text{ } rejects \text{ } w] = \sum_{i\geq 0}(1-P_a)^i(1-P_r)^iP_r = \frac{P_r}{P_a+P_r-P_aP_r} > \frac{P_r}{P_a+P_r} \tag{25}$$

$$> \frac{1/(2n^2+1)}{\varepsilon/2(n+1)^2 + 1/(2n^2+1)} = \frac{(n+1)^2/(2n^2+1)}{\varepsilon/2 + (n+1)^2/(2n^2+1)} \tag{26}$$

If we now denote $f(x) = \frac{x}{\varepsilon/2+x} = 1 - \frac{\varepsilon}{(\varepsilon+2x)}$, then $f(x)$ is monotonously increasing in $(0, +\infty)$ and since $(n+1)^2/(2n^2+1) \geq 1/2$, we have

$$Pr[(P^*, V_\varepsilon) \text{ } rejects \text{ } w] > \frac{1/2}{1/2+\varepsilon/2} = \frac{1}{1+\varepsilon} > 1-\varepsilon. \tag{27}$$

If $|w| = n$, then Steps 1 to 4.1 takes $\mathbf{O}(n)$ time, the loops 4.2 takes $\mathbf{O}(n^2)$ time, and Step 5 takes $\mathbf{O}(k)$ time. The expected number of the repetitions of the algorithm is $\mathbf{O}(2^kn^2)$ in both cases. Hence, the expected running time of (P, V_ε) is $\mathbf{O}\left(\frac{1}{\varepsilon}\left(n^4 + n^2\log\frac{1}{\varepsilon}\right)\right)$. □

Remark 3. Concerning the expected running time, if the algorithm halts with the probability $h(n) : N \to [0,1]$ in one iteration, then the expected number of repetitions till the algorithm halts is

$$\sum_{i=1}^{+\infty}(1-h(n))^{i-1}h(n) \times i = \frac{1}{h(n)}. \tag{28}$$

The halting probability in one iteration in Figure 3 is $\mathbf{\Omega}\left(\frac{1}{2^kn^2}\right)$, so according to Equality 28, the expected numbers of iterations of the algorithm is $\mathbf{O}(2^kn^2)$.

Theorem 6. *Interactive proof systems QAM(ptime-2QCFA) are more powerful than AM(ptime-2PFA).*

Proof. The fact that a coin-flipping can be simulated perfectly by 2QCFA in polynomial time, see [34], implies that 2PFA can be simulated in polynomial time by 2QCFA and therefore QAM(ptime-2QCFA) are at least as powerful as AM(ptime-2PFA). By [9], the language L_{middle} cannot be recognized by AM(ptime-2PFA) and this implies that QAM(ptime-2QCFA) are more powerful than AM(ptime-2PFA). □

3.2 Recognition of the language L_{mpal}

We show now that the language $L_{mpal} = \{xax^R \mid x \in \{a,b\}^*\}$, which cannot be recognized at all by AM(2PFA), as shown later, can be recognized by QAM(2QCFA) in an exponential expected running time.

Theorem 7. *For any ε there is a QAM(2QCFA) A_ε with a verifier-prover pair (P, V_ε) accepting the language L_{mpal} with one sided error ε in the exponential running time $\mathbf{O}\left(n\log\frac{1}{\varepsilon}\cdot 2^{n\log\frac{1}{\varepsilon}}\right)$, where $n = |w|$ is the length of the input w.*

Proof. In the QAM described in the following, the prover is used only to determine, for the verifier, the middle symbol of the input.

At the beginning of the actions of the QAM, both the verifier and the prover share an input string w. The verifier V_ε starts with the head at the left end-marker, in the initial classical state s_0 and

Repeat ad infinity:

1. Move the tape head to the right of the left end-marker.
2. If the scanned symbol is σ and the prover responds negatively to the question of the verifier whether the reading head is in the middle, then the verifier applies the operation U_σ to the state in its quantum memory and the head is moved one cell to the right and Step 2 is repeated. Otherwise, go to the next step.
3. If the scanned symbol is b (a) and the Prover informs the verifier that the head scans the symbol in the middle of the input string, the input is rejected (the head is moved to the next cell and the process goes to Step 4).
4. Until the scanned symbol σ is the right end-marker, the operation U_σ^{-1} is applied to the verifier's quantum state and the head moves one cell to the right.
5. When the right end-marker is reached, the quantum state of the verifier is measured in the basis $\{|q_0\rangle, |q_1\rangle, |q_2\rangle\}$. If the outcome is not $|q_0\rangle$, the input is rejected; otherwise go to the next step.
6. Move the head to the cell on the left from the right end-marker and set a special register *flag* to 0.
7. Until the currently scanned symbol is not the left end-marker, do the following: Simulate k coin-flips. Set *flag* to 1 if not all outcomes are "heads". Move the head one cell left.
8. If *flag* =0, accept.

Figure 4: Description of the behavior of the verifier V_ε of the QAM (P, V_ε) when the language L_{mpal} is recognized. The choice of k depends on ε, as discussed in the text.

in the initial quantum state $|q_0\rangle$ and keeps moving its head, cell by cell, till the right end-marker is reached, following the rules described informally in Figure 4, where

$$U_a = \frac{1}{5}\begin{pmatrix} 4 & 3 & 0 \\ -3 & 4 & 0 \\ 0 & 0 & 5 \end{pmatrix}, U_b = \frac{1}{5}\begin{pmatrix} 4 & 0 & 3 \\ 0 & 5 & 0 \\ -3 & 0 & 4 \end{pmatrix} \tag{29}$$

are unitary matrices that will be applied on verifier's quantum states in the Hilbert space spanned by three orthonormal basis states $|q_0\rangle, |q_1\rangle$ and $|q_2\rangle$.

An application of the unitary matrices on these basis states can also be described as follows

$$\begin{array}{l|l} U_a|q_0\rangle = \frac{4}{5}|q_0\rangle - \frac{3}{5}|q_1\rangle & U_b|q_0\rangle = \frac{4}{5}|q_0\rangle - \frac{3}{5}|q_2\rangle \\ U_a|q_1\rangle = \frac{3}{5}|q_0\rangle + \frac{4}{5}|q_1\rangle & U_b|q_1\rangle = |q_1\rangle \\ U_a|q_2\rangle = |q_2\rangle & U_b|q_2\rangle = \frac{3}{5}|q_0\rangle + \frac{4}{5}|q_2\rangle \end{array}$$

Let us now describe more formally the actions of the verifier V_ε for an input w with $|w| = n$.

Completeness condition: After the start, in each round, the verifier V_ε keeps asking the prover whether the symbol currently scanned is in the middle of the input, until he gets a positive reply. After that the verifier has no need to get more information from the prover.

Lemma 8. *If the input $w = xax^R$, then the quantum state of the verifier V_ε after Step 4 is the quantum state $|q_0\rangle$.*

Proof. Let $x = x_1x_2 \dots x_l$ for all $x_i \in \{a, b\}$. After Step 4 the quantum state will be

$$|\psi\rangle = U_{x_1}^{-1}U_{x_2}^{-1} \cdots U_{x_l}^{-1}U_{x_l} \cdots U_{x_2}U_{x_1}|q_0\rangle = |q_0\rangle. \tag{30}$$

□

Lemma 9. *An execution of Steps 6 to 8 leads to an acceptance with probability 2^{-kn}.*

Proof. The probability that at the end of Step 8 flag $= 0$ is 2^{-kn}. This is therefore the probability that the input is accepted in Step 8. If this is not the case, a new iteration starts in the state $|q_0\rangle$. □

If the input $w \in L_{mpal}$, then the verifier never rejects the input in Step 5. After Steps 6 to 8 the input is accepted with the probability $P_a = 2^{-kn}$ and is rejected with the probability $P_r = 0$. If the whole computation process is repeated indefinitely, the accepting probability is

$$Pr[(P^*, V_\varepsilon) \text{ rejects } w] = \sum_{i\geq 0}(1-P_a)^i(1-P_r)^iP_a = \frac{P_a}{P_a+P_r-P_aP_r} = \frac{P_a}{P_a} = 1. \tag{31}$$

Soundness condition: For a three dimensional vector u, let $u[i]$ denotes its i-th component.

Lemma 10. *[35] Let*

$$u = Y_1^{-1}\dots Y_l^{-1}X_m\dots X_1(1,0,0)^T$$

where $X_j, Y_j \in \{A, B\}$. If $X_j = Y_j$ for all $1 \leq j \leq m$ and $l = m$, then $u[2]^2 + u[3]^2 = 0$; otherwise $u[2]^2 + u[3]^2 > 5^{-(m+l)}$.

Lemma 11. *If the input string $w \notin L_{mpal}$, then for any prover P^* the verifier V_ε rejects w after Step 5 with the probability at least 5^{1-n}.*

Proof. Suppose that a Prover P^* informs the verifier after scanning the $(m+1)$-st symbol that it is the symbol in the middle of the input string.

If the input string w is not of the form xay with $|x| = m$, then the verifier rejects the input in Step 3. Assume now that $w = xay$, $|x| = m, |y| = l$ and $y \neq x^R$.

Let $x = x_1x_2\cdots x_m$ and $y = y_1y_2\cdots y_l$. Starting with the state $|q_0\rangle$, the verifier V_ε changes its quantum state after the loop 4 to:

$$|\psi\rangle = U_{y_l}^{-1}\cdots U_{y_2}^{-1}U_{y_1}^{-1}U_{x_m}\cdots U_{x_2}U_{x_1}|q_0\rangle. \tag{32}$$

Let $|\psi\rangle = \beta_0|q_0\rangle + \beta_1|q_1\rangle + \beta_2|q_2\rangle$. According to Lemma 10, $\beta_1^2 + \beta_2^2 > 5^{-(m+l)}$. Therefore, if in Step 5, the quantum state $|\psi\rangle$ is measured, then the input is rejected with the probability $P_r = \beta_1^2 + \beta_2^2 > 5^{-(m+l)} = 5^{1-n}$. □

If $w \notin L_{mpal}$, then, according to Lemma 11, for any Prover P^* the verifier V_ε rejects the input after Step 5 with the probability

$$P_r > 5^{1-n} \tag{33}$$

and the input is accepted after Steps 6 to 8 with the probability

$$P_a = 2^{-kn}. \tag{34}$$

If the whole computation process is repeated indefinitely, the probability that the verifier rejects the input is (taking into the consideration that $k \geq \max\{\log 5, \log\frac{1}{\varepsilon}\}$)

$$Pr[(P^*, V_\varepsilon) \text{ rejects } w] = \sum_{i\geq 0}(1-P_a)^i(1-P_r)^iP_r = \frac{P_r}{P_a+P_r-P_aP_r} \tag{35}$$

$$> \frac{P_r}{P_a+P_r} > \frac{5^{1-n}}{2^{-kn}+5^{1-n}} > \frac{1}{1+\varepsilon} > 1-\varepsilon. \tag{36}$$

Time analysis: Steps 1 to 5 take $\mathbf{O}(\text{n})$ time and Steps 6 to 8 $\mathbf{O}(kn)$ time. The halting probability is $\mathbf{\Omega}\left(\frac{1}{2^{kn}}\right)$ in both cases, so the expected number of repetitions of the above process is $\mathbf{O}(2^{kn})$ in both cases. Hence the expected running time of the QAM (P, V_ε) is $\mathbf{O}(kn \cdot 2^{kn})$ and therefore $\mathbf{O}\Big(\log\frac{1}{\varepsilon} \cdot n \cdot 2^{n \log \frac{1}{\varepsilon}}\Big)$.

□

In order to prove that the language L_{mpal} cannot be recognized by AM(2PFA) the following Lemma will be used.

Lemma 12. *[9] Let a language $L \subseteq \Sigma^*$. Suppose there is an infinite set I of positive integers such that, for each $m \in I$, there are an integer $N(m)$ and multisets $W_m = \{w_1, w_2, \cdots, w_{N(m)}\}$, $U_m = \{u_1, u_2, \cdots, u_{N(m)}\}$ and $V_m = \{v_1, v_2, \cdots, v_{N(m)}\}$ of words such that*

(1) $|w| \leq m$ for all $w \in W_m$,

(2) for every integer k there is an m_k such that $N(m) \geq m^k$ for all $m \in I$ such that $m \geq m_k$, and

(3) for all $1 \leq i, j \leq N(m)$, $u_j w_i v_j \in L$ iff $i = j$,

then L cannot be recognized by AM(2PFA).

We are now ready to prove

Theorem 13. *L_{mpal} cannot be recognized by AM(2PFA).*

Proof. Let I be the set of all positive integers. For each $m \in I$, let $N(m) = 2^m$, and let $w_1, ..., w_{N(m)}$ be an ordering of all words from $\{a, b\}^m$. Let $W_m = \{w_1, w_2, \cdots, w_{N(m)}\}$. With such a W_m, with $u_i = \lambda$ (the empty word)and $v_i = aw_i^R$ for all i, then for all $1 \leq i, j \leq N(m)$, $u_j w_i v_j \in L_{mpal}$ if and only if $i = j$. According to Lemma 12, the language L_{mpal} cannot be recognized by AM(2PFA) and theorem has been proved. □

From the last theorem and from Theorem 7 it follows:

Corollary 14. *QAM(2QCFA) are more powerful than AM(2PFA).*

3.3 Recognition of the language $L_{knapsack}$

In this subsection, we consider a language, over the alphabet $\{0, 1, \#\}$, $L_{knapsack} = \{b\#a_1\#a_2\# \cdots \#a_n \mid$ such that $a_1, \cdots, a_n, b \in 1\{0, 1\}^*,$ and there exists a set $I \subseteq \{1, \cdots, N\}$ such that $v(b) = \sum_{i \in I} v(a_i)\}$, where $v(x)$ is the number such that x is its binary representation. $L_{knapsack}$ is actually the 0 - 1 knapsack problem, which is NP-complete. Yakaryilmaz studied QAM with the verifier augmented with a fixed-size quantum register in the Arthur-Merlin proof system [32, 33] and proved that $L_{knapsack}$ can be recognized by QAM whose verifier is a 2QCFA which uses superoperators. Using an idea from [33, 32] (coding of binary strings into the amplitudes of quantum states), we prove that the language $L_{knapsack}$ can also be recognized by QAM(2QCFA). Our model of 2QCFA is weaker than that of [32, 33] and different tools, comparing those from [32, 33], are needed to prove our result.

Theorem 15. *For any $\epsilon < 1/2$, there exists a verifier-prover pair (P, V_ϵ) of a QAM(2QCFA) to recognize the language $L_{knapsack}$ with one-sided error ϵ in the exponential running time $\mathbf{O}\big(\frac{1}{\varepsilon}(n6^n + \log\frac{1}{\varepsilon})\big)$, where n is the length of the input.*

Proof. Let us assume that the input string w be of the form $b\#a_1\#a_2\#\cdots\#a_n$, where $b, a_i \in 1\{0,1\}^*$. This can be easily checked by a FA. Otherwise, the input string w is rejected immediately. The main idea of the proof is as follows: we consider a 2QCFA verifier V_ϵ that uses a quantum memory with states generated by the set of orthogonal quantum states $\{|q_i\rangle : i = 0, 1, \cdots, 7\}$, where $|q_0\rangle, |q_1\rangle, |q_2\rangle$ are used to encode the value of the binary strings. The verifier V_ϵ starts to work with the initial quantum state $|q_0\rangle$. The tape head of the verifier V_ϵ moves from the left to right. Firstly, the verifier V_ϵ encodes the value of b into the amplitudes of the quantum state $|q_1\rangle$. With the help of the prover P, the verifier V_ϵ will know whether a_i is selected or not[1]. If a_i is selected, the value of a_i is calculated and encoded into the amplitudes of the quantum state $|q_2\rangle$ and then subtracted from the amplitude of the quantum state $|q_1\rangle$. When the right end-marker \$ is reached, the verifier V_ϵ measures the current quantum state. If the resulting quantum state is $|q_1\rangle$, the input string w is rejected. Otherwise, the verifier continues as show in Figure 5, where the unitary operators and projective measurements are as follows:

```
If the input string $w$ is not of the form $b\#a_1\#a_2\#\cdots\#a_n\#$, where $b, a_i \in 1\{0,1\}^*$ (this can be easily
checked by a FA), $w$ is rejected. Otherwise, repeat the following ad infinitum:
  1. Move the tape head to the right symbol of the left end-marker.
  2. While the currently scanned symbol $\sigma$ is not '#', do the following:
     2.1 Apply $U_\sigma$ on the current quantum state, where $U_\sigma$ is defined on the text.
     2.2 Measure the current quantum state with $M_{for} = \{P_f, P_r\}$.
        2.2.1 If the measurement result is $f$, move the tape head one square to the right.
        2.2.2 Otherwise, set the quantum state to $|q_0\rangle$ and start a new iteration.
  3. While the currently scanned symbol is not the right end-marker, do the following:
     3.1 If the Prover informs the verifier that $a_i$ is not selected, move the tape head to the symbol '#'
         that is left of $a_{i+1}$.
     3.2 Otherwise, move the tape head one square to the right.
        3.2.1 While the currently scanned symbol is not '#', do the following: Apply $U'_\sigma$ on the current
              quantum state, where $\sigma$ is the currently scanned symbol. Measure the quantum state with
              $M_{for}$.
           3.2.1.1 If the measurement result is $f$, move the tape head one square to the right.
           3.2.1.2 Otherwise, set the quantum state to $|q_0\rangle$ and start a new iteration.
        3.2.2 Apply $U_\#$ on the current quantum state and then measure the quantum state with $M_{for}$.
           3.2.2.1 If the measurement result is $f$, move the tape head one square to the right.
           3.2.2.2 Otherwise, set the quantum state to $|q_0\rangle$ and start a new iteration.
  4. Measure the currently quantum state with $M_{fin}$. If the result is not $|q_0\rangle$, reject.
  5. Move the head to the cell on the left from the right end-marker and set a special register flag to 0.
  6. Until the currently scanned symbol is not the left end-marker, do the following: Simulate $k$ coin-flips.
     Set flag to 1 if not all outcomes are "heads". Move the head one cell left.
  7. If flag =0, accept.
```

Figure 5: Description of the behavior of the verifier V_ϵ of the pair (P, V_ϵ) when recognizing the language $L_{knapsack}$. The choice of k depends on ϵ, as discussed in the text.

[1] Consider the equation $v(b) = \sum_{i\in I} v(a_i) = \sum_{i=1}^{n} c_i v(a_i)$, where $c_i \in \{0,1\}$. $c_i = 1$ means a_i is selected.

$$U_0=\frac{1}{2}\begin{pmatrix} 1 & 0 & 0 & -\frac{\sqrt{6}}{2} & \frac{\sqrt{6}}{2} & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \frac{\sqrt{6}}{2} & \frac{\sqrt{6}}{2} & 0 & 0 & 0 \\ -\frac{\sqrt{6}}{2} & 0 & \frac{\sqrt{6}}{2} & -1 & 0 & 0 & 0 & 0 \\ \frac{\sqrt{6}}{2} & 0 & \frac{\sqrt{6}}{2} & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix}, U_1=\frac{1}{\sqrt{6}}\begin{pmatrix} 1 & 0 & 0 & -1 & 2 & 0 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 2 & 0 \\ 2 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & -2 & 1 & 1 \end{pmatrix}, \tag{37}$$

$$U_0'=\frac{1}{2}\begin{pmatrix} 1 & 0 & 0 & -\frac{\sqrt{6}}{2} & \frac{\sqrt{6}}{2} & 0 & 0 & 0 \\ 0 & 1 & 0 & \frac{\sqrt{6}}{2} & \frac{\sqrt{6}}{2} & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ -\frac{\sqrt{6}}{2} & \frac{\sqrt{6}}{2} & 0 & -1 & 0 & 0 & 0 & 0 \\ \frac{\sqrt{6}}{2} & \frac{\sqrt{6}}{2} & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix}, U_1'=\frac{1}{\sqrt{6}}\begin{pmatrix} 1 & 0 & 0 & -1 & 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 2 & 0 \\ 1 & 0 & 2 & 1 & 0 & 0 & 0 & 0 \\ 2 & 0 & -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -2 & -1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & -2 & 1 & 1 \end{pmatrix}, \tag{38}$$

$$U_{\#}=\begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sqrt{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sqrt{2} \end{pmatrix}, \tag{39}$$

$$M_{for}=\{P_f,P_r\} \text{ and } M_{fin}=\{P_i : i=0\cdots,7\}, \text{ where } P_f=\sum_{i=0}^{2}|q_i\rangle\langle q_i|,\ P_r=\sum_{i=3}^{7}|q_i\rangle\langle q_i| \text{ and } P_i=|q_i\rangle\langle q_i|. \tag{40}$$

Unitary operators U_0 and U_1 are used to encode the value of b, U_0' and U_1' are used to encode the value of a_i, $U_{\#}$ is used to subtract the amplitude of $|q_2\rangle$ from the amplitude of $|q_1\rangle$, respectively. The projective measurement M_{for} is used to keep the quantum state in $span\{|q_0\rangle, |q_1\rangle, |q_2\rangle\}$.

Lemma 16. *Setting the quantum state to the initial state in Step 2.2.2 can be done by a projective measurement and unitary operators.*

Proof. Let the current quantum state be $|\psi\rangle = \sum_{i=0}^{7}\alpha_i|q_i\rangle$, and the projective measurement M_{fin} be performed on $|\psi\rangle$. If the classical measurement result is i, then the resulting quantum state will be $|q_i\rangle$. By applying the unitary operator $U_i = |q_0\rangle\langle q_i|$ on the state $|q_i\rangle$, the state $|q_0\rangle$ is obtained. □

Completeness condition: If the input string $w \in L_{knapsack}$, then the all-powerful prover can make the right choice of a_i and tells that to the verifier during their communications, step by step. Let $a_{j_1}, a_{j_2}, \cdots, a_{j_s}$ be strings selected by the prover (selection is not unique) where $j_p < j_q$ if $p < q$ and $v(b) = \sum_{i=1}^{s} v(a_{j_i})$.

Lemma 17. *If the input $w \in L_{knapsack}$, then the quantum state of the verifier V_ε will be $|q_0\rangle$ after Step 4.*

Proof. Let $b = b_1 b_2 \cdots b_l$. The quantum state of the verifier V_ε after the string b is read is (see Appendix A for a detail proof)

$$|\psi_b\rangle = \frac{\prod_{i=1}^{l}(P_f U_{b_i})|q_0\rangle}{\| \prod_{i=1}^{l}(P_f U_{b_i})|q_0\rangle\|} = \frac{1}{\sqrt{1+v(b)^2}} \begin{pmatrix} 1 \\ v(b) \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{41}$$

Let $a_{j_1} = a_{j_{11}} a_{j_{12}} \cdots a_{j_{1l'}}$. The quantum state of the verifier V_ε after the string a_{j_1} is read is (see Appendix A for a detail proof)

$$|\psi_{b\#a_{j_1}}\rangle = \frac{\prod_{i=1}^{l'}(P_f U'_{a_{j_{1i}}})|\psi_b\rangle}{\| \prod_{i=1}^{l'}(P_f U'_{a_{j_{1i}}})|\psi_b\rangle\|} = \frac{1}{\sqrt{1+v(b)^2+v(a_{j_1})^2}} \begin{pmatrix} 1 \\ v(b) \\ v(a_{j_1}) \\ \vdots \\ 0 \end{pmatrix}. \tag{42}$$

Afterwards, the unitary operator $U_\#$ and the projective measurement M_{for} are performed. If the outcome of the measurement is f (that is the quantum state is in $span\{|q_0\rangle, |q_1\rangle, |q_2\rangle\}$), the resulting quantum state is

$$|\psi_{b\#a_{j_1}\#}\rangle = \frac{P_f U_\# \prod_{i=1}^{l'}(P_f U'_{a_{j_{1i}}})|\psi_b\rangle}{\|P_f U_\# \prod_{i=1}^{l'}(P_f U'_{a_{j_{1i}}})|\psi_b\rangle\|} = \frac{1}{\sqrt{1+(v(b)-v(a_{j_1}))^2}} \begin{pmatrix} 1 \\ v(b)-v(a_{j_1}) \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{43}$$

Therefore, when the verifier reaches the end of Step 4, the quantum state of the verifier is

$$|\psi_w\rangle = \frac{1}{\sqrt{1+\left(v(b)-\sum_{i=1}^{s} v(a_{j_i})\right)^2}} \begin{pmatrix} 1 \\ v(b)-\sum_{i=1}^{s} v(a_{j_i}) \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = |q_0\rangle. \tag{44}$$

□

The resulting quantum state in Step 4 will be $|q_0\rangle$ with probability 1. After Steps 5 the input is accepted with the probability $P_a = 2^{-k}$ and is rejected with the probability $P_r = 0$. If the whole process is repeated for infinitum, the accepting probability is

$$Pr[(P^*, V_\varepsilon) \text{ rejects } w] = \sum_{i\geq 0}(1-P_a)^i(1-P_r)^i P_a = \frac{P_a}{P_a+P_r-P_aP_r} = \frac{P_a}{P_a} = 1. \tag{45}$$

Soundness condition: If the input string $w \notin L_{knapsack}$ and $n = |w|$, no matter what choices the prover makes, there is not $I \subseteq \{1, \cdots, N\}$ such that $v(b) = \sum_{i \in I} v(a_i)$. Suppose $a_{j_1}, a_{j_2}, \cdots, a_{j_s}$ be the strings selected by the prover where $j_p < j_q$ if $p < q$.

Lemma 18. *If the input string $w \notin L_{knapsack}$, then for any prover P^* the verifier V_ε rejects w after Step 4 with the probability at least $\frac{1}{2}$.*

Proof. According to the analysis in Lemma 17, the quantum state at the beginning of Step 4 is

$$|\psi_w\rangle = \frac{1}{\sqrt{1 + \left(v(b) - \sum_{i=1}^{s} v(a_{j_i})\right)^2}} \begin{pmatrix} 1 \\ v(b) - \sum_{i=1}^{s} v(a_{j_i}) \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{46}$$

Obviously, $|v(b) - \sum_{i=1}^{s} v(a_{j_i})| \geq 1$. Therefore, the probability of getting the state $|q_1\rangle$ and rejecting the input w is at leat $\frac{1}{2}$. □

If $w \notin L_{knapsack}$, then, according to Lemma 18, for any prover P^* the verifier V_ε rejects the input after Step 4 with the probability

$$P_r > \frac{1}{2} \tag{47}$$

and the input is accepted after Steps 5 to 7 with the probability

$$P_a = 2^{-k}. \tag{48}$$

If the whole computation process is repeated indefinitely, the probability that the verifier rejects the input is (taking into consideration that $k = 1 + \lceil \log_2 1/\varepsilon \rceil$)

$$Pr[(P^*, V_\varepsilon) \ \text{ rejects } w] = \sum_{i \geq 0} (1 - P_a)^i (1 - P_r)^i P_r = \frac{P_r}{P_a + P_r - P_a P_r} \tag{49}$$

$$> \frac{P_r}{P_a + P_r} > \frac{1/2}{2^{-k} + 1/2} \geq \frac{1/2}{\varepsilon/2 + 1/2} = \frac{1}{1 + \varepsilon} > 1 - \varepsilon. \tag{50}$$

Time analysis: Let $n = |w|$. The probability of getting as the outcome f in the projective measurement M_{for} is at least $\frac{1}{6}$ in the whole process in Figure 5. Hence, the probability for the verifier to reach Step 4 in one iteration is at least $\left(\frac{1}{6}\right)^n$. If every outcome of the projective measures M_{for} in all computation steps in an iteration is f, then the running time from Step 1 to Step 4 is $\mathbf{O}(n)$. Therefore, the expecting running time from Step 1 to Step 4 is less than

$$\sum_{i=1}^{+\infty} \left(1 - \left(\frac{1}{6}\right)^n\right)^{i-1} \left(\frac{1}{6}\right)^n \cdot i \cdot \mathbf{O}(n) = 6^n \cdot \mathbf{O}(n). \tag{51}$$

The running time of Step 5 is $\mathbf{O}(k)$. The halting probability is $\mathbf{\Omega}\left(\frac{1}{2^k}\right)$ in both cases, so the expected number of repetitions of the algorithm is $\mathbf{O}(2^k)$ in both cases. Hence the expected running time of the QAM (P, V_ε) is $\mathbf{O}(2^k \cdot (6^n \cdot \mathbf{O}(n) + \mathbf{O}(k)))$ and therefore $\mathbf{O}\big(\frac{1}{\varepsilon}(n6^n + \log \frac{1}{\varepsilon})\big)$. □

Dwork and Stockmeyer [9] proved that the set of languages recognized by AM(2PFA) is a proper subset of P. However, in the previous theorem we prove that the language $L_{knapsack}$, which is NP complete, can be recognized by QAM(2QCFA). This is another indication that QAM(2QCFA) are more powerful than AM(2PFA).

3.4 Recognition of other languages

In this subsection, we sketch proofs for languages:

$$L_1 = \{w \mid \exists s, t, u, v \in \Sigma^*,\ w = sbt = ubv, |s| = |v|\}, \tag{52}$$

$$L_2 = \{w \mid \exists s, t, u, v \in \Sigma^*,\ w = sat = ubv, |s| = |v|\}, \tag{53}$$

that they can be recognized by QAM(2QCFA). Freivalds et al. [10] proved that the language L_1 is nonstochastic, whereas L_2, the set of nonpalindromes, is stochastic. We prove that these languages can be recognized by QAM(2QCFA) in polynomial expected running time.

Theorem 19. *For any $\varepsilon < 1/2$, there exists a verifier-prover pair (P, V_ε) of a QAM(2QCFA) that can recognize L_1 with one-sided error ε in the expected running time $\boldsymbol{O}\left(\frac{1}{\varepsilon}\left(n^4 + n^2 \log \frac{1}{\varepsilon}\right)\right)$, where n is the length of the input.*

Proof. If $w \in L_1$, then w can be of one of the following three types

Type 1. $w = sbt = ubv$, where $|s| = |u| = |v|$.

Type 2. $w = sbt = ubv$, where $|s| > |u|$. If $s = ubx$, where $x \in \Sigma^*$, then $w = ubxbt$ and $|u| = |w| - |v| - 1 = |w| - |s| - 1 = |t|$.

Type 3. $w = sbt = ubv$, where $|s| < |u|$. If $u = sby$, where $y \in \Sigma^*$, then $w = sbybv$ and $|s| = |v|$.

Therefore, $L_1 = \{w \mid \exists u, v, z \in \Sigma^*,\ w = ubv \text{ or } w = ubzbv,\ \text{where } |u| = |v|\}$.

Using the method of the proof of Theorem 1, the quantum state of the verifier will be rotated by an angle α every time a symbol in u is scanned. The prover just tells the verifier the right time to stop rotation and of which type the input is. After the verifier checks the input is of the form b or bzb, the prover tells the verifier the right position to resume rotation of the quantum state by the angle $-\alpha$. The rest of the proof is similar to the one in Theorem 1.

□

Theorem 20. *For any $\varepsilon < 1/2$, there exists a verifier-prover pair (P, V_ε) of a QAM(2QCFA) that recognizes the language L_2 with one-sided error ε in the expected running time $\boldsymbol{O}\left(\frac{1}{\varepsilon}\left(n^4 + n^2 \log \frac{1}{\varepsilon}\right)\right)$, where n is the length of input.*

Proof. Suppose $w \in L_2$. The input string w can be of one of the following types

Type 1. $w = sat = ubv$ where $|s| > |u|$. Let $s = ubx$ where $x \in \Sigma^*$, then $w = ubxat$ and $|u| = |w| - |v| - 1 = |w| - |s| - 1 = |t|$.

Type 2. $w = sat = ubv$ where $|s| < |u|$. Let $u = say$ where $y \in \Sigma^*$, then $w = saybv$ and $|s| = |v|$.

Therefore, $L_2 = \{w \mid \exists u, v, z \in \Sigma^*,\ w = uazbv \text{ or } w = ubzav,\ \text{where } |u| = |v|\}$.

To show that the proof method of Theorem 1 will be used. The prover will tell the verifier two positions – the position of the symbol after u and the position of the symbol before v. The verifier rotates its quantum state by an angle α every time it scans a symbol in u and then checks whether symbols in these two positions, pointed out by the prover, are different. If they are different, the quantum state of the verifier is rotated by an angle $-\alpha$ every time a symbol in v is scanned. Otherwise, the input string is rejected. The rest of the proof is similar to that in Theorem 1.

□

4 Concluding remarks

We have explored quantum interactive proof systems with 2QCFA verifiers and classical communication. We have focused on the public coin version of the interactive proof systems – namely QAM(2QCFA). We have showed that QAM(2QCFA) are more powerful than their classical counterparts AM(2PFA). In particular, we have shown several specific results demonstrating that: (1) The language $L_{middle} = \{xay \,|\, x, y \in \Sigma^*, \Sigma = \{a, b\}, |x| = |y|\}$ can be recognized by QAM(2QCFA) in a polynomial expected running time, but cannot be recognized by AM(2PFA) in polynomial expected running time. (2) The language $L_{mpal} = \{xax^R \,|\, x \in \{a, b\}^*\}$ can be recognized by QAM(2QCFA) in an exponential expected running time, but cannot be recognized by AM(2PFA) at all. (3) The 0 - 1 knapsack language can be recognized by QAM(2QCFA) in an exponential expected running time.

A related open problem, first mentioned in [9], is whether AM(ptime-2PFA) are more powerful than 2PFA(ptime). Our attempts (and that of others) to show that AM(ptime-2PFA) are indeed more powerful failed so far. However, results of this paper indicate, that the answer is likely positive when it comes to the quantum case. Indeed, the language L_{middle} can be recognized by QAM(ptime-2QCFA), but it seems that it can not be recognized by 2QCFA. That would imply that QAM(ptime-2QCFA) are more powerful than 2QCFA.

Acknowledgements

This work is supported in part by Employment of Newly Graduated Doctors of Science for Scientific Excellence project/grant (CZ.1.07./2.3.00/30.0009) of Czech Republic, and the National Natural Science Foundation of China (Nos. 61272058, 61073054), the Natural Science Foundation of Guangdong Province of China (No. 10251027501000004), the Specialized Research Fund for the Doctoral Program of Higher Education of China (No. 20100171110042).

References

[1] A. Ambainis, M. Beaudry, M. Golovkins, A. Kikusts, M. Mercer, D. Thénrien, Algebraic Results on Quantum Automata, Theory of Computing Systems **39** (2006) 165–188.

[2] A. Ambainis, R. Freivalds, One-way quantum finite automata: strengths, weaknesses and generalizations, in: Proceedings of the 39th Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press, Palo Alfo, California, USA, 1998, pp. 332–341.

[3] A. Ambainis, J. Watrous, Two-way finite automata with quantum and classical states, Theoretical Computer Science **287** (2002) 299–311.

[4] L. Babai, Trading group theory for randomness, in: Proc. 17th ACM Symposium on Theory of Computing, 1985, pp. 421–429.

[5] A. Bertoni, C. Mereghetti, B. Palano, Quantum Computing: 1-Way Quantum Automata, in: Proceedings of the 9th International Conference on Developments in Language Theory (DLT2003), Lecture Notes in Computer Science, Vol. 2710, Springer, Berlin, 2003, pp. 1–20.

[6] A. Brodsky, N. Pippenger, Characterizations of 1-way quantum finite automata, SIAM Journal on Computing **31** (2002) 1456–1478.

[7] A. Condon, R. Ladner, Probabilistic game automata, Journal of Computer and System Sciences **36** (3) (1988) 452–489.

[8] C. Dwork, L. Stockmeyer, A time-complexity gap for two-way probabilistic finite state automata, SIAM J. Comput. **19** (1990) 1011–1023.

[9] C. Dwork, L. Stockmeyer, Finite state verifiers I: The power of interaction, J. ACM **39** (4) (1992) 800–828.

[10] R. Freivalds, A. Yakaryilmaz, A. C. Cem Say, A new family of nonstochastic languages, Information Processing Letters **110** (2010) 410–413.

[11] S. Goldwasser, M. Sipser, Private coins versus public coins in interactive proof systems, in: Proc. 18th ACM Symposium on Theory of Computing, 1986, pp. 59–68.

[12] S. Goldwasser, S. Micali, C. Rackoff, The knowledge complexity of interactive proof systems, SIAM J. Comput. **18** (1989) 186–208.

[13] A. Greenberg, A. Weiss, A lower bound for probabilistic algorithms for finite state machines. J. Comput. System Sci. **33**(1) (1986) 88–105 .

[14] J. Gruska, Quantum Computing, McGraw-Hill, London, 1999.

[15] J. Gruska, Descriptional complexity issues in quantum computing. J. Automata, Languages Combin. **5** (3) (2000) 191–218.

[16] J. E. Hopcroft, J. D. Ullman, Introduction to Automata Theory, Languages, and Computation, Addision-Wesley, New York, 1979.

[17] A. Kondacs, J. Watrous, On the power of quantum finite state automata, in: Proceedings of the 38th IEEE Annual Symposium on Foundations of Computer Science, 1997, pp. 66–75.

[18] L. Z. Li, D. W. Qiu, X. F. Zou, L. J. Li, L. H. Wu, P. Mateus, Characterizations of one-way general quantum finite automata, Theoretical Computer Science **419** (2012) 73–91.

[19] P. Mateusa, D. W. Qiu, L. Z. Li, On the complexity of minimizing probabilistic and quantum automata, Information and Computation **218** (2012) 36–53.

[20] C. Mereghetti, B. Palano, Quantum finite automata with control language, RAIRO - Inf. Theor. Appl. **40** (2006) 315–332.

[21] C. Moore and J. P. Crutchfield, Quantum automata and quantum grammars, Theoretical Computer Science **237** (2000) 275–306.

[22] M. A. Nielsen, I. L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, Cambridge, 2000.

[23] H. Nishimura, T.Yamakami, An application of quantum finite automata to interactive proof systems. Journal of Computer and System Sciences, **75** (4) (2009) 255–269.

[24] A. Paz, Introduction to Probabilistic Automata, Academic Press, New York, 1971.

[25] D. W. Qiu, Some Observations on Two-Way Finite Automata with Quantum and Classical States, ICIC 2008, LNCS 5226 (2008), pp. 1–8.

[26] D. W. Qiu, L. Z. Li, P. Mateus, J. Gruska, Quantum finite automata, J.C. Wang, Editor, CRC Handbook of Finite State Based Models and Applications, CRC Press, 2012, pp. 113–144.

[27] Adi Shamir, IP = PSPACE. Journal of the ACM **39** (4) (1992) 869–877.

[28] J. Watrous, PSPACE has constant-round quantum interactive proof systems, Theoret. Comput. Sci. **292** (2003) 575–588.

[29] J. Watrous, Quantum computational complexity, R.A. Meyers, Editor, Encyclopedia of Complexity and Systems Science, Springer, 2009, pp. 7174–7201.

[30] A. Yakaryilmaz, A. C. Cem Say, Succinctness of two-way probabilistic and quantum finite automata, Discrete Mathematics and Theoretical Computer Science **12** (4) (2010) 19–40.

[31] A. Yakaryilmaz, A. C. Cem Say, Unbounded-error quantum computation with small space bounds, Information and Computation **209** (2011) 873–892.

[32] A. Yakaryilmaz, Turing-equivalent automata using a fixed-size quantum memory, arXiv:1205.5395 (2012).

[33] A. Yakaryilmaz, Public-qubits versus private-coins,Electronic Colloquium on Computational Complexity, Report No. 130 (2012).

[34] S. G. Zheng, D. W. Qiu, L. Z. Li, Some languages recognized by two-way finite automata with quantum and classical states, International Journal of Foundation of Computer Science, **23** (5) (2012) 1117–1129.

[35] S. G. Zheng, D. W. Qiu, J. Gruska, L. Z. Li, P. Mateus. State succinctness of two-way finite automata with quantum and classical states, arXiv:1202.2651 (2012)

Appendix A: Proofs of Equalities 41 and 42

We prove Equality 41 by induction on the length of string b. If $|b| = 1$, then $b =$ "1". Therefore,

$$U_1|q_0\rangle = \frac{1}{\sqrt{6}} \begin{pmatrix} 1 & 0 & 0 & -1 & 2 & 0 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 2 & 0 \\ 2 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & -2 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{6}} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{54}$$

and

$$|\psi_b\rangle = \frac{P_f U_1 |q_0\rangle}{\|P_f U_1 |q_0\rangle\|} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \frac{1}{\sqrt{1 + v(b)^2}} \begin{pmatrix} 1 \\ v(b) \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{55}$$

Suppose that for $b = w$ and $|w| > 1$, the equality holds, that is

$$|\psi_w\rangle = \frac{1}{\sqrt{1 + v(w)^2}} \begin{pmatrix} 1 \\ v(w) \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{56}$$

We prove the equation holds for $b = w\sigma$. If $\sigma =$ ‘$0'$, then

$$U_0|\psi_w\rangle = \frac{1}{2} \begin{pmatrix} 1 & 0 & 0 & -\frac{\sqrt{6}}{2} & \frac{\sqrt{6}}{2} & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \frac{\sqrt{6}}{2} & \frac{\sqrt{6}}{2} & 0 & 0 & 0 \\ -\frac{\sqrt{6}}{2} & 0 & \frac{\sqrt{6}}{2} & -1 & 0 & 0 & 0 & 0 \\ \frac{\sqrt{6}}{2} & 0 & \frac{\sqrt{6}}{2} & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix} \cdot \frac{1}{\sqrt{1 + v(w)^2}} \begin{pmatrix} 1 \\ v(w) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{57}$$

$$= \frac{1}{2(\sqrt{1 + v(w)^2})} \begin{pmatrix} 1 \\ 2v(w) \\ 0 \\ -\frac{\sqrt{6}}{2} \\ \frac{\sqrt{6}}{2} \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{58}$$

and

$$|\psi_{w0}\rangle = \frac{P_f U_0|\psi_w\rangle}{\|U_0|\psi_w\rangle\|} = \frac{1}{\sqrt{1+(2v(w))^2}}\begin{pmatrix} 1 \\ 2v(w) \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \frac{1}{\sqrt{1+(v(w0))^2}}\begin{pmatrix} 1 \\ v(w0) \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{59}$$

If $\sigma = `1'$, then

$$U_1|\psi_w\rangle = \frac{1}{\sqrt{6}\sqrt{1+v(w)^2}}\begin{pmatrix} 1 & 0 & 0 & -1 & 2 & 0 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 2 & 0 \\ 2 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & -2 & 1 & 1 \end{pmatrix}\begin{pmatrix} 1 \\ v(w) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{60}$$

$$= \frac{1}{\sqrt{6(1+v(w)^2)}}\begin{pmatrix} 1 \\ 2v(w)+1 \\ 0 \\ 2-v(w) \\ v(w) \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{61}$$

and

$$|\psi_{w1}\rangle = \frac{P_f U_1|\psi_w\rangle}{\|U_1|\psi_w\rangle\|} = \frac{1}{\sqrt{1+(2v(w)+1)^2}}\begin{pmatrix} 1 \\ 2v(w)+1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \frac{1}{\sqrt{1+(v(w1))^2}}\begin{pmatrix} 1 \\ v(w1) \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{62}$$

The proof of Equality 42 is similar.